

## Traitement d'image et Analyse comportementale d'Agents dans différents Environnements

En subissant les transports parisiens, je me suis rendu compte que de tels espaces publics devaient être optimisés. La disposition de l'environnement est un point crucial pour la fluidité, la sécurité, ou le confort des personnes. Le traitement d'image permet de rendre cette étude utile en l'appliquant concrètement.

Différents modèles sont étudiés pour comprendre comment peuvent interagir plusieurs agents dans un environnement qui dépend d'eux. Afin de créer un outil utile, les environnements considérés seront des environnements convertis du réel au numérique par de simples photos, puis transformés pour être étudiés.

### Positionnement thématique (ÉTAPE 1) :

- *INFORMATIQUE (Informatique pratique)*
- *MATHEMATIQUES (Mathématiques Appliquées)*

### Mots-clés (ÉTAPE 1) :

#### Mots-clés (en français) Mots-clés (en anglais)

<i>image</i>	<i>picture</i>
<i>numérisation</i>	<i>digitization</i>
<i>parcours</i>	<i>traversal</i>
<i>labyrinthe</i>	<i>maze</i>
<i>foule</i>	<i>crowd</i>

### Bibliographie commentée

Dans un premier temps, l'objectif est de numériser un environnement réel. Dans le cadre de notre étude, seules deux dimensions de cet environnement sont considérées. On le prendra en photo pour le convertir en données numériques.

L'algorithme de Canny [1] permet de détecter les contours dans une image, et donc les obstacles. Ainsi, il est possible d'effectuer des parcours en traitant l'image comme un graphe en

forme de matrice. On commence par réduire le bruit en appliquant un filtre gaussien qui floute légèrement l'image. Ensuite, on utilise le filtre de Sobel [6] pour calculer les changements d'intensité dans l'image pour chaque dimension. On ne garde ensuite que les intensités maximales locales car ce sont elles qui correspondent aux contours. Enfin, on utilise un seuillage à l'hystérésis qui permet de supprimer les hautes intensités isolées ne constituant pas de contour.

Pour l'implémenter, on pourra s'inspirer d'une première implémentation open-source [2]. Ce répertoire constitue une librairie entière qui implémente efficacement les algorithmes de Canny et de Hough en langage C. Elle servira alors d'exemple d'autant plus qu'elle inclut donc tous les sous-algorithmes nécessaires décrits au paragraphe précédent.

Pour tester le traitement d'image sur des environnements, on se restreint d'abord sur le cas particulier des labyrinthes. Ce cas sera étudié dans toute la partie traitement de l'image.

Cependant, puisque des algorithmes seront itérés de nombreuses fois sur l'image, sa qualité devient un frein bien qu'elle soit porteuse d'informations. Il est donc nécessaire de traiter l'image pour la simplifier le plus possible sans altérer ses propriétés générales. Sans cela, les résultats seront plus précis mais bien trop longs à calculer.

Le comportement de la foule peut se révéler dangereuse pour son entourage, mais surtout pour ceux qui la composent. De nombreux rassemblements ayant tourné au drame [3] constituent désormais des sujets d'étude complets afin de ne plus les reproduire [4]. Ces analyses permettent de comprendre le danger de ces bousculades, mais aussi de déterminer si un environnement est propice à leur apparition. Pour analyser la dangerosité, il est possible de mesurer la densité de personnes sur une surface, la pression subie à certains points et les variations de vitesse. Les causes sont quant à elles catégorisées en deux types, les macroscopiques et les microscopiques. Les premières correspondent aux mouvement d'ensemble tandis que les autres sont dues aux actions individuelles qui transmettent de la force comme les poussées involontaires.

Pour simuler des foules, il est d'abord nécessaire de les modéliser. Il existe de nombreuses approches [5] ce qui permet de prédire plusieurs résultats possibles. Les modèles macroscopiques regroupent les modèles de flots, s'inspirant de la mécanique des fluides, et les modèles statistiques s'intéressant directement au débit des piétons. Les modèles microscopiques quant à eux, évoluent grâce aux interactions entre individus proches en utilisant des automates cellulaires ou des forces et champs de vecteurs, se rapprochant alors de la physique.

Utiliser ces différents modèles permet de prédire différents scénarios. Il serait aussi intéressant de proposer des solutions afin d'améliorer l'environnement étudié.

## **Problématique retenue**

Le sujet a pour but de répondre aux questions suivantes : Comment numériser un environnement réel sans altérer ses propriétés ? Comment palier les défauts coûteux de cette numérisation ? Comment la simulation d'agents fictifs dans un tel environnement retranscrit-elle le comportement réel d'une foule ?

## Objectifs du TIPE du candidat

Afin de créer un outil permettant de rendre compte d'une analyse d'un environnement réel cette étude a pour objectifs :

- Étudier la conversion d'une photo d'environnement en données utiles
- Exploiter les données obtenues : être capable de simuler l'activité d'une foule dans l'environnement
- Améliorer drastiquement les performances pour une analyse accrue et une utilisation pratique en utilisant des algorithmes plus avancés
- Faire correspondre les résultats avec le réel

## Références bibliographiques (ÉTAPE 1)

- [1] PASAN KALANSOORIYA : Understanding and Implementing Canny Edge Detection in Native Python : <https://medium.com/@pasanSK/understanding-and-implementing-canny-edge-detection-in-native-python-52c296255601>
- [2] CARLOS ASENSIO MARTINEZ : Fast Edge Detection Library : <https://github.com/casensiom/fast-image-processing>
- [3] FOULOSCOPIE : Les pires mouvements de foule expliqués par la science : <https://www.youtube.com/watch?v=mhLKT4D2YvI>
- [4] WENGUO WENG, JIAYUE WANG, LIANGCHANG SHEN, YUSHAN SONG : Review of analyses on crowd-gathering risk and its evaluation methods : <https://www.sciencedirect.com/science/article/pii/S2666449622000664>
- [5] SAMUEL LEMERCIER : Simulation du comportement de suivi dans une foule de piétons à travers l'expérience, l'analyse et la modélisation : <https://theses.fr/2012REN1S028>
- [6] WIKIPEDIA : Sobel operator : [https://en.wikipedia.org/wiki/Sobel\\_operator](https://en.wikipedia.org/wiki/Sobel_operator)

## Références bibliographiques (ÉTAPE 2)

- [1] ÉQUIPE IMAGES DE TÉLÉCOM PHYSIQUE STRASBOURG : Analyse d'images, morphologie mathématique : [https://images.icube.unistra.fr/img\\_auth\\_namespace.php/6/63/3-Morphologie.pdf](https://images.icube.unistra.fr/img_auth_namespace.php/6/63/3-Morphologie.pdf)

## DOT

- [1] : *Février 2024 : Recherches bibliographiques à propos de la modélisation des floules.*
- [2] : *Avril 2024 : Premières implémentations (en Python) résultats visuels pour des environnements définis numériquement.*
- [3] : *Juillet/Août 2024 : Recherches à propos du traitement d'image.*
- [4] : *Octobre 2024 : Transition des implémentations précédentes vers le langage C par souci d'efficacité.*
- [5] : *Décembre 2024 à Février 2025 : Implémentation de la partie traitement d'image (dont filtre de Canny) et adaptation de la partie modélisation de la foule.*
- [6] : *Avril 2025 : Première version fonctionnelle du projet avec des simulations satisfaisantes.*
- [7] : *Mai 2025 : Ajout de la fermeture morphologique pour pallier la discontinuité des contours (remplaçant un épaissement naïf) et optimisation des parcours en implémentant un algorithme A\* itératif.*
- [8] : *Juin 2025 : Les simulations sont refaites avec la version la plus aboutie du projet.*